

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (Previously presented) An audio/visual system, comprising:  
at least one output component having at least one source port for each type of output signal output from the at least one output component and at least one source port object for each of said at least one source port; and  
at least one input component having at least one sink port for each type of input signal input to the at least one input component and at least one sink port object for each at least one sink port,  
wherein each at least one source port of said at least one output component is connectable to said at least one sink port of said at least one input component via at least one primitive circuit path.
2. (Previously presented) An audio/visual system according to claim 1, further including at least one primitive circuit object for each at least one primitive circuit path with a signal at least one of (A) originating from a source port and (B) terminating at a sink port.
3. (Previously presented) An audio/visual system according to claim 1, wherein the at least one primitive circuit path between the at least one source port and the at least one sink port is at least one of a static path and a dynamic path.
4. (Previously presented) An audio/visual system according to claim 1, further including at least one switching mechanism having at least one input switching port and at least one output switching port, wherein a switching mechanism of the at least one switching mechanism enables the at least one input switching port associated with the switching mechanism to be connected to the at least one output switching port associated with the switching mechanism, thereby establishing at least one dynamic path between the at least one input switching port and the at least one output switching port.
5. (Previously presented) An audio/visual system according to claim 4, wherein said at least one primitive circuit path includes at least one of (A) at least one path between at least one

source port of the at least one output component and at least one input switching port of a switching mechanism of the at least one switching mechanism and (B) at least one path between at least one output switching port of a switching mechanism of the at least one switching mechanism and at least one sink port of at least one input component.

6. (Previously presented) An audio/visual system according to claim 5, further including at least one primitive circuit object for each at least one primitive circuit path with a signal at least one of (A) originating from an output switching port and (B) terminating at an input switching port.

7. (Previously presented) An audio/visual system according to claim 4, further including at least one input switching port object for each of said at least one input switching port, and at least one output switching port object for each of said at least one output switching port.

8. (Previously presented) An audio/visual system according to claim 7, wherein said port objects, including any of said source port objects, sink port objects, input switching port objects and output switching port objects, are organized hierarchically according to at least one hierarchical level, wherein a hierarchical level includes at least one parent port object having any of (A) zero child port objects and (B) at least one child port object, wherein a port object that has no child port objects is a primitive port object and wherein a port object that is not a child port object of any other parent port object is a complete port object.

9. (Previously presented) An audio/visual system according to claim 7, wherein a parent port object of a second hierarchical level is a child port object of a first hierarchical level.

10. (Previously presented) An audio/visual system according to claim 2, wherein a virtual circuit object includes at least one reference to at least one primitive circuit object and each virtual circuit object contains primitive binding information corresponding to at least one virtual circuit path associated with the virtual circuit object.

11. (Previously presented) An audio/visual system according to claim 10, wherein each output associated with a virtual circuit object outputs a stream of signals, the signals within the stream are hierarchically organized according to how source ports are organized within a

complete source port, whereby the system represents the stream of an output component by a stream object.

12. (Previously presented) An audio/visual system according to claim 11, wherein a stream object includes at least one of (A) at least one other stream object and (B) at least one child stream object, wherein a stream object that does not contain other stream objects is a primitive stream object and a stream object that is not contained in other stream objects is a complete stream object and whereby each primitive stream object includes a signal object that corresponds to the signal that is output by at least one of (A) the corresponding source port and (B) the corresponding output switching port.

13. (Previously presented) An audio/visual system according to claim 1, wherein said at least one source port object and said at least one sink port object are derived from a port object class, and member functions of said port object class include at least one of a function that returns a reference to the owner object of at least one instance of the port object class, a function that returns an indication as to whether at least one instance of the port object class is a complete port, a function that returns an indication as to whether at least one instance of the port object class is a primitive port, a function that returns a reference to the parent port of at least one instance of the port object class, a function that returns the number of child ports of at least one instance of the port object class and a function that returns a reference to a child port for at least one instance of the port object class.

14. (Previously presented) An audio/visual system according to claim 13, wherein said at least one sink port object is of a sink port object class, derived from the port object class, and member functions of the sink port object class include at least one of a function that retains an indication as to whether at least one instance of the port object class is connected to a stream, a function that returns a reference to the stream to which the requesting sink port is connected, a function that informs a sink port that it is to consider the signals within a stream for the purpose of assigning them to a primitive sink port, a function that unassigns signals within a stream from a primitive sink port, a function that unassigns the stream from the complete sink port, a function that returns the number of assignments between a signal in the assigned stream and a primitive

sink port that are made during assignment, a function that returns an indication of the signal that is assigned to a primitive port for a given assignment number and a function that informs a complete sink port and its container of the assigned stream.

15. (Previously presented) An audio/visual system according to claim 13, wherein said at least one source port object is at least one complete source port object and is of a complete source port object class, derived from the port object class, and member functions of the complete source port object class include at least one of (A) a function that returns an indication of whether at least one instance of the complete source port object class is active, wherein a source port is active if it is producing at least one signal, (B) a function that returns a reference to the stream associated with at least one complete source port, (C) a function that returns a reference to the primitive circuit associated with at least one complete source port if the at least one complete source port is a primitive source port, (D) a function that returns the number of virtual circuits that are associated with at least one complete source port, (E) a function that returns a reference to the numbered virtual circuit, (F) a function that creates a virtual circuit that connects at least one complete source port to a designated sink port and (G) a function that removes the virtual circuit that connects at least one complete source port to a designated sink port.

16. (Previously presented) An audio/visual system according to claim 15, wherein said at least one sink port object is at least one complete sink port object and wherein a separate object at least one of enumerates, creates and removes at least one virtual circuit between a complete source port object and a complete sink port object, whereby said separate object is not said complete source port object and is not said complete sink port object.

17. (Previously presented) An audio/visual system according to claim 7, wherein said at least one input switching port object is of an input switching port object class, and member functions of said input switching port object class include at least one of (A) a function that returns the number of connections from an input switch port to output switch ports, (B) a function that returns a reference to the numbered output switch port that is connected to an input switch port, (C) a function that creates a connection from an input switch port to a designated output switch

port and (D) a function that removes a connection from an input switch port to a designated output switch port.

18. (Previously presented) An audio/visual system according to claim 7, wherein said at least one output switching port object is of an output switching port object class, and a member function of said output switching port object class is a function that retrieves the input switch port to which the output switch port is connected.

19. (Previously presented) An audio/visual system according to claim 2, wherein said at least one primitive circuit object is of a primitive circuit object class, and member functions of said primitive circuit object class include at least one of a function that returns a reference to the primitive source port of a primitive circuit and a function that returns a reference to the primitive sink port of a primitive circuit.

20. (Previously presented) An audio/visual system according to claim 10, wherein said virtual circuit object is of a virtual circuit object class, and member functions of said virtual circuit object class include at least one of a function that returns a reference to a complete source port that is producing the signals being routed by the subject virtual circuit, a function that returns a reference to a complete sink port that is receiving the signals being routed by the virtual circuit, a function that returns the number of bindings between primitive source ports and primitive sink ports for the designated virtual connection and a function that returns the designated numbered binding as a reference to the associated primitive source port and a reference to the associated primitive sink port.

21. (Previously presented) An audio/visual system according to claim 11, wherein signals within a stream are of a signal class, wherein member functions of the signal class include at least one of (A) a function that returns the intended usage of the requesting signal, (B) a function that returns the format of the requesting signal, (C) a function that returns a reference to the stream which is the parent of the signal and (D) a function that returns a reference to the primitive source port that is outputting the signal.

22. (Previously presented) An audio/visual system according to claim 21, wherein member functions include properties of the signal class.

23. (Previously presented) An audio/visual system according to claim 11, wherein a stream is of a stream class, wherein member functions of the stream class includes at least one of a function that enables the enumeration of at least one child stream and a function that enables the retrieval of at least one child stream.

24. (Previously presented) An audio/visual system according to claim 23, wherein a stream is of a stream class, wherein member functions of the stream class include at least one of (A) a function that returns an indication as to whether a stream is a complete stream, (B) a function that returns an indication as to whether a stream is a primitive stream, (C) a function that returns a reference to the stream that is the parent of a stream, (D) a function that returns the number of child streams of a stream, (E) a function that returns a reference to the designated numbered child stream of a stream, (F) a function that returns a reference to the source port that is producing a stream, (G) a function that returns a reference to the source program that is producing a stream and (H) a function that returns a reference to a signal in a stream.

25. (Previously presented) An audio/visual system according to claim 1, wherein a wire protocol-based approach is utilized that supports the semantics of said at least one output component and said at least one input component.

26. (Previously presented) An audio/visual system according to claim 1, further including a switching mechanism that connects said at least one source port of said at least one output component to said at least one sink port of said at least one input component and wherein the connecting by said switching mechanism includes dynamically allocating a stream transport resource.

27. (Previously presented) An audio/visual system according to claim 26, wherein said switching mechanism is one of a software mechanism and a hardware mechanism.

28. (Previously presented) An audio/visual system according to claim 26, wherein said switching mechanism is a hierarchically nested switching mechanism.

29. (Previously presented) An audio/visual system according to claim 26, wherein the connecting by said switching mechanism of said at least one source port of said at least one

output component to said at least one sink port of said at least one input component includes allocating a stream transport resource in accordance with at least one type of at least one signal being transported between said at least one source port and said at least one sink port at the time of said allocating.

30. (Previously presented) An audio/visual system according to claim 29, wherein said allocating includes generating at least one virtual circuit that matches at least one capability of said at least one source port and at least one capability of said at least one sink port.

31. (Previously presented) An audio/visual system according to claim 26, wherein said switching mechanism includes a third party object that requests the generating of at least one virtual circuit for connecting said at least one source port of said at least one output component to said at least one sink port of said at least one input component.

32. (Previously presented) An audio/visual system according to claim 1, further comprising:  
at least one entertainment session; and  
at least one player/recorder component associated with each entertainment session including at least one output component, wherein a player/recorder component is a type of source object.

33. (Previously presented) An audio/visual system according to claim 32, wherein said at least one player/recorder component is one of a tuner, a tape machine, a compact disk player, a laser disk player, a removable disk drive, a hard drive, a video cassette recorder, a digital versatile disk player, a video game system and a computing device.

34. (Previously presented) An audio/visual system according to claim 32, wherein an entertainment session of said at least one entertainment session provides a behavior that allows an audio/visual program to be assigned to a player/recorder component, and whereby when an audio/visual program is assigned to an entertainment session, the entertainment session loads the audio/visual program into a player/recorder, causes the program to be played by the player/recorder and routes at least one output signal of the player/recorder component to at least one associated output component.

35. (Previously presented) An audio/visual system according to claim 34, wherein said loading of the audio/visual program into a player/recorder by said entertainment session includes at least one of (A) instantiating the audio/visual program on a player/recorder and (B) resolving the audio/visual program to a player/recorder appropriate for the audio/visual program.

36. (Previously presented) An audio/visual system according to claim 1, wherein said at least one output component includes at least one of (A) a display component and (B) a speaker system component, wherein a display object represents a display component and a speaker system object represents a speaker system component.

37. (Previously presented) An audio/visual system according to claim 36, wherein a speaker system includes a sink port and wherein a display component includes a sink port.

38. (Previously presented) An audio/visual system according to claim 34, wherein a space object is associated with each entertainment session designating its space, a player/recorder object is associated with each player/recorder component.

39. (Previously presented) An audio/visual system according to claim 34, wherein an entertainment session includes at least one default output component, such that when an audio/visual program is assigned to the entertainment session, the at least one output signal for the player/recorder component is routed to at least one default output component.

40. (Previously presented) An audio/visual system according to claim 34, wherein an entertainment session creates at least one virtual circuit to route said audio/visual program from said player/recorder component and to said at least one output component.

41. (Previously presented) An audio/visual system according to claim 34, wherein an entertainment session at least one of dynamically creates at least one virtual circuit to route said audio/visual program to a plurality of output components and dynamically destroys at least one existing virtual circuit no longer needed to route said audio/visual program.

42. (Previously presented) An audio/visual system according to claim 34, wherein said entertainment session provides said behavior in response to an external action in said system.



43. (Previously presented) An audio/visual system according to claim 34, wherein, for each of its associated output components, an entertainment session at least one of (A) determines whether the routing of the audio/visual program is possible, (B) is notified of an action external to the entertainment session and (C) determines whether to provide a user interface for controlling the at least one output component to which the at least one signal is routed.

44. (Previously presented) An audio/visual system according to claim 43, wherein when the entertainment session is notified that one of its output components has been activated due to an external action, the entertainment session becomes an additional controller of the player/recorder component outputting to the output component.

45. (Previously presented) An audio/visual system according to claim 43, wherein said action external to the entertainment session includes a physical load of an audio/visual program into a physical device.

46. (Previously presented) An audio/visual system according to claim 34, wherein an entertainment session provides a property notification when a property of one of an associated player/recorder component and associated output component changes.

47. (Previously presented) An audio/visual system according to claim 46, wherein said providing of a property notification includes notifying a user interface component corresponding to the at least one player/recorder component and output component.

48. (Previously presented) An audio/visual system according to claim 34, wherein an entertainment session provides a user interface component for controlling at least one user interface of at least one of (A) at least one input component and (B) at least one output component associated with the entertainment session.

49. (Previously presented) An audio/visual system according to claim 34, wherein a player/recorder object has at least one associated complete source port object and has at least one associated complete sink port object and wherein each output component has at least one associated complete sink port and wherein the player/recorder object provides a behavior at least one of (A) to load an audio/visual program into a player/recorder component, (B) to allow at

least one command to be sent to the player/recorder component, (C) to determine whether it is possible to load an audio/visual program into the player/recorder component and (D) to provide custom behavior customized to the corresponding player/recorder component.

50. (Previously presented) An audio/visual system according to claim 34, wherein an output component has a type and an output component provides at least one of (A) a behavior that returns the identification of a sink port object that is appropriate for assigning the signals to the output component for a specified stream object, (B) a behavior that is specific to the type of output component, wherein the behavior is one of (a) part of the base object class and (b) provided through a derivation of that base object class.

51. (Previously presented) An audio/visual system according to claim 34, further comprising:

a program pool data structure hierarchically representing a set of audio/visual program entries, whereby each audio/visual program entry has a corresponding program pool data structure.

52. (Previously presented) An audio/visual system according to claim 51, wherein an audio/visual program entry includes an audio/visual program object.

53. (Previously presented) An audio/visual system according to claim 51, wherein an audio visual program entry includes a program type and program types include (A) broadcast program (B) single access physical media and (C) multiple access media.

54. (Previously presented) An audio/visual system according to claim 53, wherein an audio/visual program entry includes a program type, and wherein possible program types include a tuner type, a tape machine type, a compact disk player type, a laser disk player type, a removable disk drive type, a hard drive type, a video cassette recorder type, a digital versatile disk player type, a video game system type, a JPEG image type, a streaming media type and a computing device type.

55. (Previously presented) An audio/visual system according to claim 34, wherein an audio/visual program entry includes at least one child program object, and wherein said at least one child program object is hierarchically organized.

56. (Previously presented) An audio/visual system according to claim 34, wherein a program pool data structure at least one of (A) provides a behavior to browse through the hierarchy of the audio/visual program entries represented by the program pool data structure, (B) allows a player/recorder component to be assigned to an audio/visual program entry of the program pool data structure, (C) provides a behavior corresponding to the loading of an audio/visual program entry into a player/recorder component and (D) allows an entertainment session to be created by a session manager.

57. (Previously presented) An audio/visual system according to claim 56, wherein said providing of a behavior corresponding to the loading of an audio/visual program entry includes at least one of (A) instantiating the audio/visual program corresponding to the audio/visual program entry on a player/recorder and (B) resolving the audio/visual program corresponding to the audio/visual program entry to a player/recorder appropriate for the audio/visual program.

58. (Previously presented) An audio/visual system according to claim 34, wherein an audio/visual program entry includes the identifier of an owner of the audio/visual program entry.

59. (Previously presented) An audio/visual system according to claim 58, wherein the owner is one of (A) a second audio/visual program entry and (B) the program pool data structure that includes the audio/visual program entry.

60. (Previously presented) An audio/visual system according to claim 59, wherein an audio/visual program entry at least one of (A) allows for the retrieving of its child program objects, (B) allows for the retrieving of its parent program object of which the audio/visual program entry is a child program object and (C) enables the establishment of at least one criterion so that only child program objects that match the at least one criterion are returned.

61. (Previously presented) An audio/visual system according to claim 60, wherein a parent program object of an audio/visual program entry is retrieved via the associated program pool

data structure by providing the location of the audio/visual program entry to the program pool data structure.

62. (Previously presented) An audio/visual system according to claim 34, further comprising:

a program pool data structure hierarchically representing a set of audio/visual program entries, whereby each audio/visual program entry has a corresponding program pool data structure.

63. (Previously presented) An audio/visual system according to claim 62, wherein an audio/visual program entry includes a program identification (ID), which provides descriptive information about the audio/visual program represented by the audio/visual program entry.

64. (Previously presented) An audio/visual system according to claim 63, wherein said descriptive information includes at least one of a name associated with the audio/visual program, a time associated with the audio/visual program, a volume associated with the audio/visual program, a genre associated with the audio/visual program and a format associated with the audio/visual program.

65. (Previously presented) An audio/visual system according to claim 64, wherein said descriptive information includes at least one location of at least one medium that corresponds to the audio/visual program.

66. (Previously presented) An audio/visual system according to claim 65, wherein a location is represented as a path within a hierarchy of locations.

67. (Previously presented) An audio/visual system according to claim 63, wherein an audio/visual program entry has an associated program type, which specifies a path through a hierarchy of program types.

68. (Previously presented) An audio/visual system according to claim 63, wherein functions of components of the audio/visual system resolve a program ID into a plurality of different types of references including (A) a get program object function that resolves a program ID into a

reference to a corresponding audio/visual program entry, (B) a get program genre function that resolves a program ID into a plurality of references to a set of audio/visual program entries in the same genre.

69. (Previously presented) An audio/visual system according to claim 68, wherein the get program genre function causes the audio/visual program entry associated with the program ID to retrieve information relating its genre.

70. (Previously presented) An audio/visual system according to claim 34, wherein an audio/visual program entry exposes an interface for maintenance of state of the audio/visual program entry including at least one of (A) an interface for one of adding and deleting a property of the audio/visual program entry, (B) an interface for setting a property of the audio/visual program entry, (C) an interface for one of adding and deleting a child program object of the audio/visual program entry and (D) an interface for deleting of the audio/visual program entry itself.

71. (Previously presented) An audio/visual system according to claim 70, wherein an interface is specific to the type of audio/visual program represented by the audio/visual program entry.

72. (Previously presented) An audio/visual system according to claim 34, wherein a program pool data structure provides an access port for each client that is accessing the program pool and wherein the program pool data structure exposes a function that receives a program ID and returns a reference to an audio/visual program entry corresponding to that program ID.

73. (Previously presented) An audio/visual system according to claim 34, wherein a program pool data structure allows for database cursor-like access to the program objects, whereby when a query is submitted to the program pool data structure which specifies at least one criterion for audio/visual program entries, the program objects of the program pool data structure that match the at least one criterion are provided in a result set accessible to the client.

74. (Previously presented) An audio/visual system according to claim 73, wherein a partial list of program objects that match the at least one criterion are provided in a result set while query continues to be processed.

75. (Previously presented) An audio/visual system according to claim 73, wherein a client accesses the result set using at least one of (A) a function to advance to the next program object in the result set, (B) a get reference function for the current program object which returns a reference to the current program object of the result set and (C) a return a set of references function for the program objects in the result set that returns a set of references to the program objects.

76. (Previously presented) An audio/visual system according to claim 73, wherein the result set of a query is cached at a client and wherein the program pool data structure automatically updates the client's cache as the set of programs that match the at least one criterion changes.

77. (Previously presented) An audio/visual system according to claim 75, wherein the program pool data structure provides an access control mechanism to restrict access by an enumerated client.

78. (Previously presented) An audio/visual system according to claim 1, further comprising:  
at least one media manager object for managing media at its location and for providing at least one appropriate object, as determined by the at least one media manager object, for the media managed by the at least one media manager object.

79. (Previously presented) An audio/visual system according to claim 78, wherein said at least one appropriate object is at least one player/recorder object.

80. (Previously presented) An audio/visual system according to claim 78, wherein given an audio/visual program, a media manager object provides automatic resolution of at least one appropriate device for the audio/visual program.

81. (Previously presented) An audio/visual system according to claim 80, wherein said at least one appropriate device is at least one source port.

82. (Previously presented) An audio/visual system according to claim 78, wherein a media manager object includes a load program function that is passed an audio/visual program entry and that returns at least one appropriate object with the program corresponding to the audio/visual program entry loaded.

83. (Previously presented) An audio/visual system according to claim 82, wherein the load function results in dynamic allocation of an appropriate source device for the program corresponding to the audio/visual program entry.

84. (Previously presented) An audio/visual system according to claim 82, wherein the load function results in dynamic allocation of an appropriate source device based on mapping attributes of the audio/visual program entry, wherein said mapping attributes include an indication of location of the program corresponding to the audio/visual program entry.

85. (Previously presented) An audio/visual system according to claim 82, wherein the at least one media manager object is hierarchically organized according to parent media manager objects and child media manager objects where a child media manager object exists.

86. (Previously presented) An audio/visual system according to claim 85, wherein each parent media manager object of at least one child media manager object includes the ability to map the location of an audio/visual program to the media manager object that is responsible for returning the player/recorder object for the audio/visual program entry corresponding to the audio/visual program.

87. (Previously presented) An audio/visual system according to claim 86, wherein each parent media manager object of at least one child media manager object includes an associated location table, which maps the location of an audio/visual program to the media manager object that is responsible for returning the player/recorder object for the audio/visual program entry corresponding to the audio/visual program.

88. (Previously presented) An audio/visual system according to claim 85, wherein a media manager object that has no child media manager object processes the location of the audio/visual program entry to identify which player/recorder to associate with the audio/visual program entry.

89. (Previously presented) An audio/visual system according to claim 32, wherein the process of assigning of a program to an entertainment session includes:

invoking a function to select an audio/visual program entry thereby returning a reference to the audio/visual program entry; and

invoking a set current program function of the entertainment session object passing the reference to the audio/visual program entry.

90. (Previously presented) An audio/visual system according to claim 32, wherein the process of selecting a program from an entertainment session includes:

displaying a user interface that allows a user to browse through the programs associated with a program pool data structure;

selecting a program via the user interface; and

setting a return reference to a reference to a program object associated with the selected program.

91. (Previously presented) An audio/visual system according to claim 32, wherein the process of selecting a program from an entertainment session includes:

inputting a program monitored by the entertainment session; and

automatically selecting by the entertainment session a program object related to the input program.

92. (Previously presented) An audio/visual system according to claim 91, wherein the input program is a physical medium.

93. (Previously presented) An audio/visual system according to claim 90, wherein the user interface allows the user to specify a search criterion.

94. (Previously presented) An audio/visual system according to claim 32, wherein invoking the set current program function of an entertainment session object passes a reference to an



audio/visual program entry thereby loading that audio/visual program entry within the entertainment session.

95. (Previously presented) An audio/visual system according to claim 94, wherein said invoking includes:

- invoking a function to retrieve a loaded player/recorder object;
- passing a reference to the audio/visual program entry; and
- returning a reference to a player/recorder object that is loaded with the program.

96. (Previously presented) An audio/visual system according to claim 95, further including:

- invoking a get current source function of the player/recorder object, thereby returning a reference to a complete source port for the player/recorder object; and
- invoking a get stream reference function of the source port object to retrieve a reference to a complete stream for the source port object.

97. (Previously presented) An audio/visual system according to claim 96, further including:

- looping while selecting at least one output component associated with the entertainment session; and
- creating a virtual circuit from the player/recorder component to each of the output components.

98. (Previously presented) An audio/visual system according to claim 97, wherein said looping includes at least one of synchronous looping and asynchronous looping.

99. (Previously presented) An audio/visual system according to claim 97, wherein said looping includes:

- requesting a selected output component to return a sink port object that is appropriate to the type of stream;
- invoking a get sink port function of the output object corresponding to the selected output component; and
- invoking a create virtual circuit function of the source port object passing a reference to the sink port object, thereby creating a virtual circuit from the source port to the sink port.

100. (Previously presented) An audio/visual system according to claim 32, wherein the entertainment session includes a load program function to retrieve a loaded player/recorder object, which passes a reference to an audio/visual program entry and returns a reference that has been allocated to a player/recorder object, said load program includes:

retrieving the location information from the audio/visual program entry;

if the location information indicates that a player/recorder component is associated with the audio/visual program entry, invoking a load function of the audio/visual program entry and receiving a reference to a loaded player/recorder object in return; and

if the location information indicates that a player/recorder component is not already associated with the audio/visual program entry, receiving a reference to the loaded player/recorder object.

101. (Previously presented) An audio/visual system according to claim 100, further including:

retrieving a player/recorder object that is appropriate to an associated entertainment session; and

invoking a load program function of the player/recorder object passing the reference to the audio/visual program entry.

102. (Previously presented) An audio/visual system according to claim 101, wherein the load function of the player/recorder object includes:

invoking a load program function of the media manager object passing a reference to the audio/visual program entry, receiving a reference to a player/recorder object in return; and

invoking the load program function of the player/recorder object passing the program reference.

103. (Previously presented) An audio/visual system according to claim 100, wherein the load program function of a player/recorder object, which is passed a reference to an audio/visual program entry and effects the loading of the program into that player/recorder component includes:

identifying a complete source port of the player/recorder component that is appropriate for the passed program;

assigning the audio/visual program entry to the player/recorder object;  
determining at least one of the usage, format and port type for the primitive ports of the selected source port;  
invoking a set signal function of the complete source port passing said at least one of the usage, format and port type, thereby setting the usage, format and port type for each primitive source port; and  
notifying the audio/visual program entry that it has been loaded.

104. (Previously presented) An audio/visual system according to claim 102, wherein the load program function of a media manager object, which is performed when the media manager object has at least one child media manager object, includes:

passing a reference to an audio/visual program entry and returning a reference to a player/recorder object;  
invoking a get location function of the audio/visual program entry to retrieve location information from the audio/visual program entry;  
searching a location table for a media manager object that manages the media corresponding to the audio/visual program entry; and  
invoking the load program function of the located media manager object.

105. (Previously presented) An audio/visual system according to claim 102, wherein the load program function of a media manager object, which is performed when the media manager object has zero child media manager objects, includes:

retrieving location information from the audio/visual program entry and automatically finding the media associated with the location information;  
initializing an appropriate object for the media; and  
setting a return reference to the appropriate object.

106. (Previously presented) A method for establishing a path between a source component and an input component in an audio/visual system comprising at least one source component having at least one source port capable of supporting each type of output signal output from the at least one source component and at least one primitive source port object for each of said at

least one source port; and at least one input component having at least one sink port capable of supporting each type of input signal input to the at least one input component and a primitive sink port object for each sink port, wherein each at least one source port of said at least one source component is connectable to said at least one sink port of said at least one input component via at least one primitive circuit path, including:

instantiating a virtual circuit object that establishes a path between the source port corresponding to the complete source port object and the sink port corresponding to the complete sink port object.

107. (Previously presented) A method according to claim 106, further including:  
establishing a primitive circuit path using an output object that represents the source component and an input object that represents the input component.

108. (Previously presented) A method according to claim 106, further including:  
requesting the output object to provide a reference to a complete source port object;  
requesting the complete source port object to provide a reference to its corresponding complete stream object; and  
requesting the input object to provide a reference to its corresponding complete sink port object.

109. (Previously presented) A method according to claim 106, wherein the sink port and the source port are included in the same port.

110. (Previously presented) A method according to claim 106, wherein said instantiating includes hosting the virtual circuit object by the source port.

111. (Previously presented) A method according to claim 106, wherein said method further includes:

invoking a create virtual circuit function wherein said invoking includes passing a reference to the sink port object;

in response to said invoking, constructing a new virtual circuit object wherein said constructing includes passing to a constructor a reference to the source port object and a reference to the sink port object; and

adding the new virtual circuit object to a list of virtual circuits associated with the source port object.

112. (Previously presented) A method according to claim 111, wherein said constructing of the new virtual circuit object includes:

retrieving by the constructor a reference to the stream associated with the source port object;

assigning the stream by the constructor to the sink port object by invoking an assign stream function of the sink port object passing a reference to the stream object, thereby returning the number of signal objects within the stream object that are assigned to the complete sink port object; and

creating a primitive binding object for each signal object that is assigned to the sink port object.

113. (Previously presented) A method according to claim 112, wherein said creating of a primitive binding object for each signal object includes:

selecting by the constructor the first signal number, and if the first signal number has been selected, selecting by the constructor the next signal number;

if the selected number is less than or equal to the number of assigned signals, retrieving by the constructor (A) a reference to the primitive sink port object corresponding to the numbered signal object and (B) a reference to the signal object, wherein said retrieving includes invoking a get assignment reference function of the sink port object;

retrieving by the constructor a reference to the primitive source port object for the corresponding signal port object.

retrieving by the constructor a reference to the sink port object of the primitive source port object;

if the primitive sink port object of the primitive circuit of the primitive sink port object is the same as the primitive sink port object of the primitive circuit of the primitive source port

object, directly connecting the source port and the sink port, otherwise, connecting the source port and the sink port through a switching mechanism.

114. (Previously presented) A method according to claim 113, wherein if the connection is through a switching mechanism, invoking by the constructor a process-not-direct-connection function; and

adding by the constructor an identification of a binding from the primitive source port to the primitive sink port to a binding table of the virtual circuit object, wherein the binding represents the identity of the primitive source port object, the identity of the input switch port object of the switching mechanism, the identity of the output switch port object of the switching mechanism and the identity of the primitive sink port object.

115. (Previously presented) A method according to claim 114, wherein said invoking of the process-not-direct-connection function includes:

retrieving a reference to the switch input port object for the primitive circuit of the primitive source port object;

retrieving a reference to the primitive source port object;

retrieving a reference to the output switch port object of the retrieved primitive circuit;

and

creating a connection between the input switch port object and the output switch port object.

116. (Previously presented) A method according to claim 113, wherein if the connection is direct,

adding by the constructor an identification of a binding from the primitive source port to the primitive sink port to a binding table of the virtual circuit object, wherein the binding represents the identity of the primitive source port object and the identity of the primitive sink port object.

117. (Previously presented) A computing device comprising means for performing the method of claim 106.

118. (Previously presented) A computing device comprising means for performing the method of claim 106, wherein said means for performing includes hardware.

119. (Previously presented) At least one computer readable medium comprising computer executable modules having computer executable instructions for performing the method of claim 106.

120-124 (Cancelled)

125. (Previously presented) A method for use in connection with an audio/visual system, comprising:

connecting at least one source port of at least one output component to at least one sink port of at least one input component via at least one primitive circuit path;

wherein each of said at least one output component includes at least one source port for each type of output signal output from the at least one output component and at least one source port object for each of said at least one source port, and

wherein each of said at least one input component includes at least one sink port for each type of input signal input to the at least one input component and at least one sink port object for each at least one sink port.

126. (Previously presented) A method according to claim 125, wherein at least one primitive circuit object is generated for each at least one primitive circuit path with a signal at least one of (A) originating from a source port and (B) terminating at a sink port.

127. (Previously presented) A method according to claim 125, wherein the at least one primitive circuit path between the at least one source port and the at least one sink port is at least one of a static path and a dynamic path.

128. (Previously presented) A method according to claim 125, wherein a switching mechanism having at least one input switching port and at least one output switching port connects at least one input switching port associated with the switching mechanism to the at least one output switching port associated with the switching mechanism, thereby establishing at least

one dynamic path between the at least one input switching port and the at least one output switching port.

129. (Previously presented) A method according to claim 128, wherein said at least one primitive circuit path includes at least one of (A) at least one path between at least one source port of the at least one output component and at least one input switching port of a switching mechanism of the at least one switching mechanism and (B) at least one path between at least one output switching port of a switching mechanism of the at least one switching mechanism and at least one sink port of at least one input component.

130. (Previously presented) A method according to claim 129, further including generating at least one primitive circuit object for each at least one primitive circuit path with a signal at least one of (A) originating from an output switching port and (B) terminating at an input switching port.

131. (Previously presented) A method according to claim 128, further including generating at least one input switching port object for each of said at least one input switching port, and generating at least one output switching port object for each of said at least one output switching port.

132. (Previously presented) A method according to claim 131, wherein said port objects, including any of said source port objects, sink port objects, input switching port objects and output switching port objects, are organized hierarchically according to at least one hierarchical level, wherein a hierarchical level includes at least one parent port object having any of (A) zero child port objects and (B) at least one child port object, wherein a port object that has no child port objects is a primitive port object and wherein a port object that is not a child port object of any other parent port object is a complete port object.

133. (Previously presented) A method according to claim 131, wherein a parent port object of a second hierarchical level is a child port object of a first hierarchical level.

134. (Previously presented) A method according to claim 126, wherein a virtual circuit object includes at least one reference to at least one primitive circuit object and each virtual circuit



object contains primitive binding information corresponding to at least one virtual circuit path associated with the virtual circuit object.

135. (Previously presented) A method according to claim 134, further including outputting a stream of signals by each output associated with a virtual circuit object, wherein the signals within the stream are hierarchically organized according to how source ports are organized within a complete source port, whereby the system represents the stream of an output component by a stream object.

136. (Previously presented) A method according to claim 135, wherein a stream object includes at least one of (A) at least one other stream object and (B) at least one child stream object, wherein a stream object that does not contain other stream objects is a primitive stream object and a stream object that is not contained in other stream objects is a complete stream object and whereby each primitive stream object includes a signal object that corresponds to the signal that is output by at least one of (A) the corresponding source port and (B) the corresponding output switching port.

137. (Previously presented) A method according to claim 125, wherein said at least one source port object and said at least one sink port object are derived from a port object class, and member functions of said port object class include at least one of a function that returns a reference to the owner object of at least one instance of the port object class, a function that returns an indication as to whether at least one instance of the port object class is a complete port, a function that returns an indication as to whether at least one instance of the port object class is a primitive port, a function that returns a reference to the parent port of at least one instance of the port object class, a function that returns the number of child ports of at least one instance of the port object class and a function that returns a reference to a child port for at least one instance of the port object class.

138. (Previously presented) A method according to claim 137, wherein said at least one sink port object is of a sink port object class, derived from the port object class, and member functions of the sink port object class include at least one of a function that retains an indication as to whether at least one instance of the port object class is connected to a stream, a function

that returns a reference to the stream to which the requesting sink port is connected, a function that informs a sink port that it is to consider the signals within a stream for the purpose of assigning them to a primitive sink port, a function that unassigns signals within a stream from a primitive sink port, a function that unassigns the stream from the complete sink port, a function that returns the number of assignments between a signal in the assigned stream and a primitive sink port that are made during assignment, a function that returns an indication of the signal that is assigned to a primitive port for a given assignment number and a function that informs a complete sink port and its container of the assigned stream.

139. (Previously presented) A method according to claim 137, wherein said at least one source port object is at least one complete source port object and is of a complete source port object class, derived from the port object class, and member functions of the complete source port object class include at least one of (A) a function that returns an indication of whether at least one instance of the complete source port object class is active, wherein a source port is active if it is producing at least one signal, (B) a function that returns a reference to the stream associated with at least one complete source port, (C) a function that returns a reference to the primitive circuit associated with at least one complete source port if the at least one complete source port is a primitive source port, (D) a function that returns the number of virtual circuits that are associated with at least one complete source port, (E) a function that returns a reference to the numbered virtual circuit, (F) a function that creates a virtual circuit that connects at least one complete source port to a designated sink port and (G) a function that removes the virtual circuit that connects at least one complete source port to a designated sink port.

140. (Previously presented) A method according to claim 139, wherein said at least one sink port object is at least one complete sink port object and wherein a separate object at least one of enumerates, creates and removes at least one virtual circuit between a complete source port object and a complete sink port object, whereby said separate object is not said complete source port object and is not said complete sink port object.

141. (Previously presented) A method according to claim 131, wherein said at least one input switching port object is of an input switching port object class, and member functions of said

input switching port object class include at least one of (A) a function that returns the number of connections from an input switch port to output switch ports, (B) a function that returns a reference to the numbered output switch port that is connected to an input switch port, (C) a function that creates a connection from an input switch port to a designated output switch port and (D) a function that removes a connection from an input switch port to a designated output switch port.

142. (Previously presented) A method according to claim 131, wherein said at least one output switching port object is of an output switching port object class, and a member function of said output switching port object class is a function that retrieves the input switch port to which the output switch port is connected.

143. (Previously presented) A method according to claim 126, wherein said at least one primitive circuit object is of a primitive circuit object class, and member functions of said primitive circuit object class include at least one of a function that returns a reference to the primitive source port of a primitive circuit and a function that returns a reference to the primitive sink port of a primitive circuit.

144. (Previously presented) A method according to claim 134, wherein said virtual circuit object is of a virtual circuit object class, and member functions of said virtual circuit object class include at least one of a function that returns a reference to a complete source port that is producing the signals being routed by the subject virtual circuit, a function that returns a reference to a complete sink port that is receiving the signals being routed by the virtual circuit, a function that returns the number of bindings between primitive source ports and primitive sink ports for the designated virtual connection and a function that returns the designated numbered binding as a reference to the associated primitive source port and a reference to the associated primitive sink port.

145. (Previously presented) A method according to claim 135, wherein signals within a stream are of a signal class, wherein member functions of the signal class include at least one of (A) a function that returns the intended usage of the requesting signal, (B) a function that returns the format of the requesting signal, (C) a function that returns a reference to the stream which is

the parent of the signal and (D) a function that returns a reference to the primitive source port that is outputting the signal.

146. (Previously presented) A method according to claim 145, wherein member functions include properties of the signal class.

147. (Previously presented) A method according to claim 135, wherein a stream is of a stream class, wherein member functions of the stream class includes at least one of a function that enables the enumeration of at least one child stream and a function that enables the retrieval of at least one child stream.

148. (Previously presented) A method according to claim 147, wherein a stream is of a stream class, wherein member functions of the stream class include at least one of (A) a function that returns an indication as to whether a stream is a complete stream, (B) a function that returns an indication as to whether a stream is a primitive stream, (C) a function that returns a reference to the stream that is the parent of a stream, (D) a function that returns the number of child streams of a stream, (E) a function that returns a reference to the designated numbered child stream of a stream, (F) a function that returns a reference to the source port that is producing a stream, (G) a function that returns a reference to the source program that is producing a stream and (H) a function that returns a reference to a signal in a stream.

149. (Previously presented) A method according to claim 125, wherein a wire protocol-based approach is utilized that supports the semantics of said at least one output component and said at least one input component.

150. (Previously presented) A method according to claim 125, further including connecting by a switching mechanism said at least one source port of said at least one output component to said at least one sink port of said at least one input component and wherein the connecting by said switching mechanism includes dynamically allocating a stream transport resource.

151. (Previously presented) A method according to claim 150, wherein said switching mechanism is one of a software mechanism and a hardware mechanism.

152. (Previously presented) A method according to claim 150, wherein said switching mechanism is a hierarchically nested switching mechanism.

153. (Previously presented) A method according to claim 150, wherein the connecting by said switching mechanism of said at least one source port of said at least one output component to said at least one sink port of said at least one input component includes allocating a stream transport resource in accordance with at least one type of at least one signal being transported between said at least one source port and said at least one sink port at the time of said allocating.

154. (Previously presented) A method according to claim 153, wherein said allocating includes generating at least one virtual circuit that matches at least one capability of said at least one source port and at least one capability of said at least one sink port.

155. (Previously presented) A method according to claim 150, wherein said switching mechanism includes a third party object that requests the generating of at least one virtual circuit for connecting said at least one source port of said at least one output component to said at least one sink port of said at least one input component.

156. (Previously presented) A method according to claim 125, further comprising:  
generating at least one entertainment session; and  
associating at least one player/recorder component with each entertainment session including at least one output component, wherein a player/recorder component is a type of source object.

157. (Previously presented) A method according to claim 156, wherein said at least one player/recorder component is one of a tuner, a tape machine, a compact disk player, a laser disk player, a removable disk drive, a hard drive, a video cassette recorder, a digital versatile disk player, a video game system and a computing device.

158. (Previously presented) A method according to claim 156, further including providing a behavior by an entertainment session of said at least one entertainment session, wherein the behavior allows an audio/visual program to be assigned to a player/recorder component, and whereby when an audio/visual program is assigned to an entertainment session, further

comprising loading the audio/visual program into a player/recorder by the entertainment session, causing the program to be played by the player/recorder and routing at least one output signal of the player/recorder component to at least one associated output component.

159. (Previously presented) A method according to claim 158, wherein said loading of the audio/visual program into a player/recorder by said entertainment session includes at least one of (A) instantiating the audio/visual program on a player/recorder and (B) resolving the audio/visual program to a player/recorder appropriate for the audio/visual program.

160. (Previously presented) A method according to claim 125, wherein said at least one output component includes at least one of (A) a display component and (B) a speaker system component, wherein a display object represents a display component and a speaker system object represents a speaker system component.

161. (Previously presented) A method according to claim 160, wherein a speaker system includes a sink port and wherein a display component includes a sink port.

162. (Previously presented) A method according to claim 158, wherein a space object is associated with each entertainment session designating its space, a player/recorder object is associated with each player/recorder component.

163. (Previously presented) A method according to claim 158, wherein an entertainment session includes at least one default output component, such that when an audio/visual program is assigned to the entertainment session, the at least one output signal for the player/recorder component is routed to at least one default output component.

164. (Previously presented) A method according to claim 158, further comprising creating by an entertainment session at least one virtual circuit to route said audio/visual program from said player/recorder component and to said at least one output component.

165. (Previously presented) A method according to claim 158, further comprising at least one of dynamically creating by an entertainment session at least one virtual circuit to route said

audio/visual program to a plurality of output components and dynamically destroying at least one existing virtual circuit no longer needed to route said audio/visual program.

166. (Previously presented) A method according to claim 158, wherein said entertainment session provides said behavior in response to an external action in said system.

167. (Previously presented) A method according to claim 158, further comprising, for each of its associated output components, at least one of (A) determining by an entertainment session whether the routing of the audio/visual program is possible, (B) notifying an entertainment session of an action external to the entertainment session and (C) determining by an entertainment session whether to provide a user interface for controlling the at least one output component to which the at least one signal is routed.

168. (Previously presented) A method according to claim 167, further including the entertainment session becoming an additional controller of a player/recorder component outputting to the output component when the entertainment session is notified that one of its output components has been activated due to an external action.

169. (Previously presented) A method according to claim 167, wherein said action external to the entertainment session includes a physical load of an audio/visual program into a physical device.

170. (Previously presented) A method according to claim 158, further including providing by an entertainment session a property notification when a property of one of an associated player/recorder component and associated output component changes.

171. (Previously presented) A method according to claim 170, wherein said providing of a property notification includes notifying a user interface component corresponding to the at least one player/recorder component and output component.

172. (Previously presented) A method according to claim 158, further including providing by an entertainment session a user interface component for controlling at least one user interface of

at least one of (A) at least one input component and (B) at least one output component associated with the entertainment session.

173. (Previously presented) A method according to claim 158, wherein a player/recorder object has at least one associated complete source port object and has at least one associated complete sink port object and wherein each output component has at least one associated complete sink port and further comprising providing a behavior by the player/recorder object at least one of (A) to load an audio/visual program into a player/recorder component, (B) to allow at least one command to be sent to the player/recorder component, (C) to determine whether it is possible to load an audio/visual program into the player/recorder component and (D) to provide custom behavior customized to the corresponding player/recorder component.

174. (Previously presented) A method according to claim 158, wherein an output component has a type and an output component provides at least one of (A) a behavior that returns the identification of a sink port object that is appropriate for assigning the signals to the output component for a specified stream object, (B) a behavior that is specific to the type of output component, wherein the behavior is one of (a) part of the base object class and (b) provided through a derivation of that base object class.

175. (Previously presented) A method according to claim 158, further comprising:  
generating a program pool data structure hierarchically representing a set of audio/visual program entries, whereby each audio/visual program entry has a corresponding program pool data structure.

176. (Previously presented) A method according to claim 175, wherein an audio/visual program entry includes an audio/visual program object.

177. (Previously presented) A method according to claim 175, wherein an audio/visual program entry includes a program type and program types include (A) broadcast program (B) single access physical media and (C) multiple access media.

178. (Previously presented) A method according to claim 177, wherein an audio/visual program entry includes a program type, and wherein possible program types include a tuner type,



a tape machine type, a compact disk player type, a laser disk player type, a removable disk drive type, a hard drive type, a video cassette recorder type, a digital versatile disk player type, a video game system type, a JPEG image type, a streaming media type and a computing device type.

179. (Previously presented) A method according to claim 158, wherein an audio/visual program entry includes at least one child program object, and wherein said at least one child program object is hierarchically organized.

180. (Previously presented) A method according to claim 158, further comprising at least one of (A) providing by a program pool data structure a behavior to browse through the hierarchy of the audio/visual program entries represented by the program pool data structure, (B) allowing by a program pool data structure a player/recorder component to be assigned to an audio/visual program entry of the program pool data structure, (C) providing by a program pool data structure a behavior corresponding to the loading of an audio/visual program entry into a player/recorder component and (D) allowing by a program pool data structure an entertainment session to be created by a session manager.

181. (Previously presented) A method according to claim 180, wherein said providing of a behavior corresponding to the loading of an audio/visual program entry includes at least one of (A) instantiating the audio/visual program corresponding to the audio/visual program entry on a player/recorder and (B) resolving the audio/visual program corresponding to the audio/visual program entry to a player/recorder appropriate for the audio/visual program.

182. (Previously presented) A method according to claim 158, wherein an audio/visual program entry includes the identifier of an owner of the audio/visual program entry.

183. (Previously presented) A method according to claim 182, wherein the owner is one of (A) a second audio/visual program entry and (B) the program pool data structure that includes the audio/visual program entry.

184. (Previously presented) A method according to claim 60, further comprising at least one of (A) allowing by an audio/visual program entry for the retrieving of its child program objects, (B) allowing by an audio/visual program entry for the retrieving of its parent program object of

which the audio/visual program entry is a child program object and (C) enabling by an audio/visual program entry the establishment of at least one criterion so that only child program objects that match the at least one criterion are returned.

185. (Previously presented) A method according to claim 184, further including retrieving by a parent program object an audio/visual program entry, via the associated program pool data structure by providing the location of the audio/visual program entry to the program pool data structure.

186. (Previously presented) A method according to claim 158, further comprising:  
generating a program pool data structure hierarchically representing a set of audio/visual program entries, whereby each audio/visual program entry has a corresponding program pool data structure.

187. (Previously presented) A method according to claim 186, wherein an audio/visual program entry includes a program identification (ID), which provides descriptive information about the audio/visual program represented by the audio/visual program entry.

188. (Previously presented) A method according to claim 187, wherein said descriptive information includes at least one of a name associated with the audio/visual program, a time associated with the audio/visual program, a volume associated with the audio/visual program, a genre associated with the audio/visual program and a format associated with the audio/visual program.

189. (Previously presented) A method according to claim 188, wherein said descriptive information includes at least one location of at least one medium that corresponds to the audio/visual program.

190. (Previously presented) A method according to claim 189, wherein a location is represented as a path within a hierarchy of locations.

191. (Previously presented) A method according to claim 187, wherein an audio/visual program entry has an associated program type, which specifies a path through a hierarchy of program types.

192. (Previously presented) A method according to claim 187, wherein functions of components of the audio/visual system resolve a program ID into a plurality of different types of references including (A) a get program object function that resolves a program ID into a reference to a corresponding audio/visual program entry, (B) a get program genre function that resolves a program ID into a plurality of references to a set of audio/visual program entries in the same genre.

193. (Previously presented) A method according to claim 192, further including causing via the get program genre function the audio/visual program entry associated with the program ID to retrieve information relating its genre.

194. (Previously presented) A method according to claim 158, further including exposing by an audio/visual program entry an interface for maintenance of state of the audio/visual program entry including at least one of (A) an interface for one of adding and deleting a property of the audio/visual program entry, (B) an interface for setting a property of the audio/visual program entry, (C) an interface for one of adding and deleting a child program object of the audio/visual program entry and (D) an interface for deleting of the audio/visual program entry itself.

195. (Previously presented) A method according to claim 194, wherein an interface is specific to the type of audio/visual program represented by the audio/visual program entry.

196. (Previously presented) A method according to claim 158, further including providing by a program pool data structure an access port for each client that is accessing the program pool and wherein the program pool data structure exposes a function that receives a program ID and returns a reference to an audio/visual program entry corresponding to that program ID.

197. (Previously presented) A method according to claim 158, further comprising providing by a program pool data structure for database cursor-like access to the program objects, and providing the program objects of the program pool data structure that match the at least one

criterion in a result set accessible to the client when a query is submitted to the program pool data structure which specifies at least one criterion for audio/visual program entries.

198. (Previously presented) A method according to claim 197, wherein said providing includes providing a partial list of program objects that match the at least one criterion in a result set while query continues to be processed.

199. (Previously presented) A method according to claim 197, further comprising accessing by a client the result set using at least one of (A) a function to advance to the next program object in the result set, (B) a get reference function for the current program object which returns a reference to the current program object of the result set and (C) a return a set of references function for the program objects in the result set that returns a set of references to the program objects.

200. (Previously presented) A method according to claim 197, further comprising caching the result set of a query at a client and automatically updating by the program pool data structure the client's cache as the set of programs that match the at least one criterion changes.

201. (Previously presented) A method according to claim 200, wherein said accessing includes providing by the program pool data structure an access control mechanism to restrict access by an enumerated client.

202. (Previously presented) A method according to claim 125, further comprising:  
generating at least one media manager object for managing media at its location and for providing at least one appropriate object, as determined by the at least one media manager object, for the media managed by the at least one media manager object.

203. (Previously presented) A method according to claim 202, wherein said at least one appropriate object is at least one player/recorder object.

204. (Previously presented) A method according to claim 202, further comprising, given an audio/visual program, providing automatic resolution by a media manager object of at least one appropriate device for the audio/visual program.

205. (Previously presented) A method according to claim 204, wherein said at least one appropriate device is at least one source port.

206. (Previously presented) A method according to claim 202, wherein a media manager object includes a load program function that is passed an audio/visual program entry and that returns at least one appropriate object with the program corresponding to the audio/visual program entry loaded.

207. (Previously presented) A method according to claim 206, wherein the load function results in dynamic allocation of an appropriate source device for the program corresponding to the audio/visual program entry.

208. (Previously presented) A method according to claim 206, wherein the load function results in dynamic allocation of an appropriate source device based on mapping attributes of the audio/visual program entry, wherein said mapping attributes include an indication of location of the program corresponding to the audio/visual program entry.

209. (Previously presented) A method according to claim 206, wherein the at least one media manager object is hierarchically organized according to parent media manager objects and child media manager objects where a child media manager object exists.

210. (Previously presented) A method according to claim 209, wherein each parent media manager object of at least one child media manager object includes the ability to map the location of an audio/visual program to the media manager object that is responsible for returning the player/recorder object for the audio/visual program entry corresponding to the audio/visual program.

211. (Previously presented) A method according to claim 210, wherein each parent media manager object of at least one child media manager object includes an associated location table, which maps the location of an audio/visual program to the media manager object that is responsible for returning the player/recorder object for the audio/visual program entry corresponding to the audio/visual program.

212. (Previously presented) A method according to claim 209, wherein a media manager object that has no child media manager object processes the location of the audio/visual program entry to identify which player/recorder to associate with the audio/visual program entry.

213. (Previously presented) A method according to claim 156, wherein the process of assigning of a program to an entertainment session includes:

- invoking a function to select an audio/visual program entry thereby returning a reference to the audio/visual program entry; and

- invoking a set current program function of the entertainment session object passing the reference to the audio/visual program entry.

214. (Previously presented) A method according to claim 156, wherein the process of selecting a program from an entertainment session includes:

- displaying a user interface that allows a user to browse through the programs associated with a program pool data structure;

- selecting a program via the user interface; and

- setting a return reference to a reference to a program object associated with the selected program.

215. (Previously presented) A method according to claim 156, wherein the process of selecting a program from an entertainment session includes:

- inputting a program monitored by the entertainment session; and

- automatically selecting by the entertainment session a program object related to the input program.

216. (Previously presented) A method according to claim 215, wherein the input program is a physical medium.

217. (Previously presented) A method according to claim 214, wherein the user interface allows the user to specify a search criterion.

218. (Previously presented) A method according to claim 156, wherein invoking the set current program function of an entertainment session object passes a reference to an audio/visual program entry thereby loading that audio/visual program entry within the entertainment session.

219. (Previously presented) A method according to claim 218, wherein said invoking includes:

- invoking a function to retrieve a loaded player/recorder object;
- passing a reference to the audio/visual program entry; and
- returning a reference to a player/recorder object that is loaded with the program.

220. (Previously presented) A method according to claim 219, further including:

- invoking a get current source function of the player/recorder object, thereby returning a reference to a complete source port for the player/recorder object; and

- invoking a get stream reference function of the source port object to retrieve a reference to a complete stream for the source port object.

221. (Previously presented) A method according to claim 220, further including:

- looping while selecting at least one output component associated with the entertainment session; and

- creating a virtual circuit from the player/recorder component to each of the output components.

222. (Previously presented) A method according to claim 221, wherein said looping includes at least one of synchronous looping and asynchronous looping.

223. (Previously presented) A method according to claim 221, wherein said looping includes: requesting a selected output component to return a sink port object that is appropriate to the type of stream;

- invoking a get sink port function of the output object corresponding to the selected output component; and

- invoking a create virtual circuit function of the source port object passing a reference to the sink port object, thereby creating a virtual circuit from the source port to the sink port.

224. (Previously presented) A method according to claim 156, wherein the entertainment session includes a load program function to retrieve a loaded player/recorder object, which passes a reference to an audio/visual program entry and returns a reference that has been allocated to a player/recorder object, said load program includes:

- retrieving the location information from the audio/visual program entry;

- if the location information indicates that a player/recorder component is associated with the audio/visual program entry, invoking a load function of the audio/visual program entry and receiving a reference to a loaded player/recorder object in return; and

- if the location information indicates that a player/recorder component is not already associated with the audio/visual program entry, receiving a reference to the loaded player/recorder object.

225. (Previously presented) A method according to claim 224, further including:

- retrieving a player/recorder object that is appropriate to an associated entertainment session; and

- invoking a load program function of the player/recorder object passing the reference to the audio/visual program entry.

226. (Previously presented) A method according to claim 225, wherein the load function of the player/recorder object includes:

- invoking a load program function of the media manager object passing a reference to the audio/visual program entry, receiving a reference to a player/recorder object in return; and

- invoking the load program function of the player/recorder object passing the program reference.

227. (Previously presented) A method according to claim 224, wherein the load program function of a player/recorder object, which is passed a reference to an audio/visual program entry and effects the loading of the program into that player/recorder component, includes:

- identifying a complete source port of the player/recorder component that is appropriate for the passed program;

- assigning the audio/visual program entry to the player/recorder object;



determining at least one of the usage, format and port type for the primitive ports of the selected source port;

invoking a set signal function of the complete source port passing said at least one of the usage, format and port type, thereby setting the usage, format and port type for each primitive source port; and

notifying the audio/visual program entry that it has been loaded.

228. (Previously presented) A method according to claim 226, wherein the load program function of a media manager object, which is performed when the media manager object has at least one child media manager object, includes:

passing a reference to an audio/visual program entry and returning a reference to a player/recorder object;

invoking a get location function of the audio/visual program entry to retrieve location information from the audio/visual program entry;

searching a location table for a media manager object that manages the media corresponding to the audio/visual program entry; and

invoking the load program function of the located media manager object.

229. (Previously presented) A method according to claim 226, wherein the load program function of a media manager object, which is performed when the media manager object has zero child media manager objects, includes:

retrieving location information from the audio/visual program entry and automatically finding the media associated with the location information;

initializing an appropriate object for the media; and

setting a return reference to the appropriate object.

230. (Previously presented) At least one of an operating system, driver code, an application programming interface, a tool kit and a processing device for providing the method of claim 125.

231. (Previously presented) A modulated data signal carrying computer executable instructions for performing the method of claim 125.

232. (Previously presented) A computing device comprising means for performing the method of claim 125.

233-243 (Cancelled)